

AD-A128 397

FORTTRAN IMPLEMENTATION OF COMPLEX LEAST SQUARES
ADAPTIVE LATTICE ALGORITHM..(U) PENNSYLVANIA STATE UNIV
UNIVERSITY PARK APPLIED RESEARCH LAB..
B A COOPER ET AL. 12 NOV 82

1/0

UNCLASSIFIED

F/G 12/1

NL

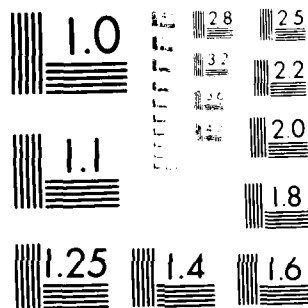
END

DATE

FILED

6-83

DTIC



MICROCOPY RESOLUTION TEST CHART
 NATIONAL BUREAU OF STANDARDS-1963-A

AD A 128337

FORTRAN IMPLEMENTATION OF COMPLEX LEAST SQUARES
ADAPTIVE LATTICE ALGORITHM AS AN ALL ZERO INVERSE
FILTER

B. A. Cooper and L. H. Sibul

Technical Memorandum
File No. TM 82-219
November 12, 1982
Contract No. N00024-79-C-6043

Copy No. 20

Copy available to DTIC does not
permit fully legible reproduction

The Pennsylvania State University
Intercollege Research Programs and Facilities
APPLIED RESEARCH LABORATORY
Post Office Box 30
State College, PA 16801

Approved for Public Release
Distribution Unlimited

NAVY DEPARTMENT
NAVAL SEA SYSTEMS COMMAND

DTIC FILE COPY

83

05

23

002

DTIC
ELECTE
MAY 23 1983
S D
E

DISCLAIMER NOTICE

**THIS DOCUMENT IS BEST QUALITY
PRACTICABLE. THE COPY FURNISHED
TO DTIC CONTAINED A SIGNIFICANT
NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.**

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TM 82-219	2. GOVT ACCESSION NO. AD 82 397	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) FORTRAN IMPLEMENTATION OF COMPLEX LEAST SQUARES ADAPTIVE LATTICE ALGORITHM AS AN ALL ZERO INVERSE FILTER		5. TYPE OF REPORT & PERIOD COVERED Interim
7. AUTHOR(s) B. A. Cooper and L. H. Sibul		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Applied Research Laboratory P. O. Box 30 State College, PA 16801		8. CONTRACT OR GRANT NUMBER(s) N00024-79-C-6043
11. CONTROLLING OFFICE NAME AND ADDRESS D. Houser, Code 63R-14 Naval Sea Systems Command Department of the Navy, Washington, DC 20362		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE November 12, 1982
		13. NUMBER OF PAGES 51
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release, distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) fortran, adaptive, algorithm, speech, signal processing		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A complex least squares adaptive lattice algorithm (1) has been implemented on the ARL VAX/VMS computer as an inverse all zero filter. A FORTRAN program package is presented which also includes a recursive realization of a pole-zero prefilter and an input signal generator. A series of examples is taken from the speech field, using the inverse filter to deconvolve speech signals produced by vocal tract models of varying complexity. The input to the prefilter is shown to be accurately recovered as the output of the inverse filter.		

DD FORM 1473

JAN 73

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	



Subject: FORTRAN Implementation of Complex Least Squares Adaptive Lattice Algorithm as an All Zero Inverse Filter

References: See Pages 14 and 15.

ABSTRACT

A complex least squares adaptive lattice algorithm (1) has been implemented on the ARL VAX/VMS computer as an inverse all zero filter. A FORTRAN program package is presented which also includes a recursive realization of a pole-zero prefilter and an input signal generator. A series of examples is taken from the speech field, using the inverse filter to deconvolve speech signals produced by vocal tract models of varying complexity. The input to the prefilter is shown to be accurately recovered as the output of the inverse filter.

TABLE OF CONTENTS

	<u>Page No.</u>
ABSTRACT	1
TABLE OF CONTENTS	2
LIST OF TABLES	3
LIST OF FIGURES	3
INTRODUCTION	4
BACKGROUND	4
PROGRAM PACKAGE	7
GEN.FOR	9
FILTER.FOR	10
LSL.FOR	12
ACKNOWLEDGMENTS	14
REFERENCES	14-15
APPENDIX A	26-45
APPENDIX B	46-51

LIST OF TABLES

<u>Table No.</u>		<u>Page No.</u>
I	RELATIONSHIPS BETWEEN LATTICE PARAMETERS, REF. 1 SYMBOLS, AND FORTRAN VARIABLES	23-24
II	FORMANT FREQUENCIES, THEIR BANDWIDTHS, AND CORRESPONDING FILTER COEFFICIENTS FOR THE FOUR INPUT CASES DEPICTED IN FIGS. 3-6 ...	25

LIST OF FIGURES

<u>Figure No.</u>		<u>Page No.</u>
1	LATTICE (INVERSE) FILTER STRUCTURE	16
2	BLOCK FLOW DIAGRAM OF LATTICE (INVERSE) FILTER IMPLEMENTATION	17
3	EACH STATE OF THE SYSTEM (SEE FIG. 2) FOR AN IMPULSE FUNCTION INPUT TO THE ONE- FORMANT CASE	18
4	EACH STATE OF THE SYSTEM (SEE FIG. 2) FOR A NOISE INPUT TO THE ONE-FORMANT CASE	19
5	EACH STATE OF THE SYSTEM (SEE FIG. 2) FOR AN IMPULSE FUNCTION INPUT TO THE FIVE- FORMANT CASE	20
6	EACH STATE OF THE SYSTEM (SEE FIG. 2) FOR A NOISE INPUT TO THE FIVE-FORMANT CASE ...	21
7	CALLING HIERARCHY AND STRUCTURE OF COMMAND PROCEDURE	22

INTRODUCTION

A complex least squares lattice adaptive algorithm (1) has been implemented on the ARL VAX/VMS computer as an inverse all zero filter. An interactive FORTRAN program package is presented which also includes a recursive realization of a pole-zero prefilter and an input signal generator. No knowledge of FORTRAN is required to use the package. The input signal characteristics, filter coefficients and critical lattice parameters are all user specified.

Three classes of adaptive algorithms are treated in the literature: (Widrow's) LMS, the gradient lattice, and the least squares lattice. Of the three, the least squares adaptive lattice algorithm has been shown to have the fastest convergence rate (2,3). For this reason, it has been used successfully in a wide range of applications; i.e., data deconvolution, system identification, speech processing, and spectral whitening. The program package described in this memorandum was developed with these applications in mind.

BACKGROUND

Consider a filter with transfer function

$$H(z) = \frac{Z(z)}{V(z)} = \frac{B(z)}{A(z)} \quad (1)$$

where

$$A(z) = 1 + \sum_{k=1}^{NPOLE} A_k z^{-k} = \sum_{k=0}^{NPOLE} A_k z^{-k}, \quad A_0 = 1$$

and

$$B(z) = 1 + \sum_{k=1}^{NZERO} B_k z^{-k} = \sum_{k=0}^{NZERO} B_k z^{-k}, B_0 = 1$$

then $Z(z)A(z) = B(z)V(z)$ and, after taking inverse z transforms of both sides, the filter can be realized in the time domain by the following recursion:

$$z(n) + \sum_{k=1}^{NPOLE} A_k z(n-k) = v(n) + \sum_{k=1}^{NZERO} B_k v(n-k)$$

or

$$z(n) = - \sum_{k=1}^{NPOLE} A_k z(n-k) + v(n) + \sum_{k=1}^{NZERO} B_k v(n-k).$$

The roots of $A(z)$ are the poles of the system; the roots of $B(z)$ are the zeroes of the system. For stability, these roots must lie outside the unit circle because they are roots of polynomials in z^{-1} .

The resonances and antiresonances of the human vocal tract (for a particular steady state vowel sound) can be modeled by poles and zeroes, respectively, in the z plane. It is a common practice in the speech field to model these formants with an all pole model rather than with poles and zeroes; adding a sufficient number of poles has been shown to closely approximate the effect of the zeroes. The roots of the denominator polynomial are related to the vocal tract formants by the following expression (5):

$$z_k, z_k^* = z_{2k}, z_{2k-1} = e^{-\sigma_k T} e^{\pm j 2\pi F_k T}$$

where F_k is the formant frequency and $2\sigma_k$ is the bandwidth of the k^{th} formant. There will be a pair of complex conjugate poles for each formant frequency; the order of the system will be twice the number of formants.

Then

$$H(z) = \frac{1}{\prod_{k=1}^{NPOLE} (1 + z_k z^{-1})}.$$

For the all-pole model, the time domain recursion becomes

$$z(n) = -\sum_{k=1}^{NPOLE} A_k z(n-k) + v(n).$$

An inverse filter for the system of Equation 1 would have the transfer function

$$H(z) = \frac{A(z)}{B(z)}.$$

For the speech case (all-pole prefilter), the inverse filter would be all-zero:

$$H_{\text{pre}}(z) = \frac{1}{A(z)}; H_{\text{inv}}(z) = \frac{1}{H_{\text{pre}}(z)} = A(z).$$

Obviously, if the systems are cascaded, the input to the prefilter will be recovered as the output of the inverse filter.

A complex adaptive lattice structure has been used to develop the inverse filter. The actual lattice algorithm is taken from (1). Users are referred to (2) and (3) and the references cited therein for the derivations of the complex least squares adaptive lattice algorithm.

PROGRAM PACKAGE

The lattice program package consists of the following interactive
FORTRAN programs:

GEN.FOR	input generator
FILTER.FOR	prefilter and realization of the lattice filter
LSL.FOR	complex adaptive lattice processor

the following subroutines:

SYNTH	synthesizes complex polynomial from formants and bandwidths
SYNTH1	synthesizes complex polynomial from its roots
ZCPOLY	solves polynomial with complex coefficients

and the following functions:

MTH\$RANDOM	(VAX/VMS library) uniform random number generator
AMAXLOC	finds location of highest value in an array
ARRMAX	finds highest value in an array

The execution of the programs may be controlled by either one of
two command procedures: LATTICE.COM when using a VT100 terminal, or
LATTEK.COM when using the Tektronix terminal in Room 358, ASB. These
command procedures are given in Appendix A. They may be invoked by
typing @LATTICE and @LATTEK, respectively. The following command should
be placed into either the command procedure or the user's LOGIN.COM file
to allow access to the IMSL package. If LNK\$LIBRARY has already been
assigned, use LNK\$LIBRARY1 or the next highest unassigned library.

\$ASSIGN DRA0:[SYSTEM]IMSLIBS.OLB LNK\$LIBRARY

Appendix B presents a sample terminal session for the case illustrated in Figure 3. For the user interested in modifying the programs, the rest of this memorandum deals with the structure of the individual programs.

Figure 7 illustrates the calling hierarchy of the lattice package. The inputs to program LSL are Gaussian white noise and the output time series $z(n)$ from program FILTER.FOR. The outputs from LSL are the set of inverse filter coefficients (A_k ; $k=1, \dots, \text{NZERO}$) to which the lattice has adapted after the requested number of iterations, and the time series output $e_p(n)$ which is equivalent to the result of passing the input time series $z(n)$ through the transfer function $H(z) = A(z)$. The output filter coefficients are fed back into program FILTER to be realized recursively with input $z(n)$. Thus, since program FILTER operates for any $H(z)$, it is both a prefilter and the realization of the lattice filter.

A sample case, then, may be illustrated by a series of five plots (Figures 3-6). The relationships between the plots are shown by the block diagram in Figure 2:

- (a) input time series
- (b) prefilter transfer function
- (c) output of the prefilter
- (d) lattice (inverse) filter transfer function
(realized by program FILTER)
- (e) output of the lattice filter

So that the filter outputs may be most easily evaluated, the filter outputs (c) and (e) for deterministic prefilter inputs are time series plots; for a stochastic input, the outputs (c) and (e) are power spectra. The time series plots (c) and (e) (from above) for the deterministic case are

presented in Figures 3-6 for one period of the waveform after the time series has stabilized (or adapted). Again, the output of the lattice filter should approximate the input to the prefilter if the lattice is a true inverse filter. It is obvious that some measure of goodness of the inverse filter should be applied to the output of the lattice filter. At present, only a graphical comparison has been attempted. However, Gray and Markel (7) suggest a spectral flatness measure which will help to smooth the discontinuities in the power spectrum caused by the occurrence of the inverse filter zeroes being offset by a few Hz. from the prefilter poles. Also, a mean square error criterion might be applied to the difference between the deterministic time series output (error signal) and the input time series. The transfer function of the lattice filter should be the inverse of the prefilter transfer function. Thus, where the prefilter has poles (peaks) the inverse filter will have zeroes (dips).

GEN.FOR

Program GEN.FOR generates both Gaussian white noise and an impulse sequence. A more accurate approximation to the glottal wave is given in (7). However, since the thrust of this work is aimed at demonstrating that the lattice filter is indeed an accurate inverse filter, the two cases used so far are sufficient models of voiced and unvoiced steady state vocal tract excitations.

Program GEN.FOR calls the uniform random number generator, MTHSRANDOM, from the VAX/VMS library. The Gaussian noise is then developed following a technique from (9). The output time series are in complex format but the imaginary parts are zero. The sampling frequency is 10 kHz. For

the complex impulse input, the frequency is variable; the examples shown in Figures 3 and 5 use $F_0 = 125$ Hz, a reasonable fundamental frequency for an adult male voice. The interactive user inputs to program GEN are F_0 , fundamental frequency; M , number of desired samples; and σ , the standard deviation of the noise. The time series are each stored in an output disk file.

FILTER.FOR

The user selects the desired type of input and the source of the filter coefficients. Several options exist concerning specification of coefficients. The formant frequencies and their bandwidths may be specified and the coefficients calculated from subroutine SYNTH and then stored on disk. The stored coefficients may be used with another input option, reading in coefficients from disk. The coefficients may be typed in directly from the keyboard. For the realization of the lattice (inverse) filter, the forward predictor (outputs from program LSL) coefficients may be read in from disk as the lattice filter coefficients. The other interactive user inputs to FILTER.FOR are M , desired number of time samples; $NPOLE$, number of poles; and $NZERO$, the number of zeroes. It is possible to specify a prefilter transfer function which has both poles and zeroes (the inverse filter would still be all zero). The modeling of a pole zero process by an all-pole process is discussed by Friedlander and Maitra (6). The examples presented here involve prefilters containing only poles for the purpose of illustrating the accuracy of the all-zero inverse filter.

The filter gain is computed by the following formula:

$$G = 10 \text{ LOG } \frac{\sum_{n=1}^M z^2(n)}{\sum_{n=1}^M v^2(n)} = \frac{\sigma_{out}^2}{\sigma_{in}^2}$$

where $u = 0$ and $\sigma_z^2 = \sigma_x^2 + \sigma_y^2$.

The filter transfer function is realized from the filter coefficients by the following formula:

$$20 \text{ LOG } |H(\omega)| = 20 \text{ LOG } \left| \frac{1}{A(z)} \right| = 20 \text{ LOG } \left[\frac{1}{A(z)} \cdot \frac{1}{A^*(z)} \right]^{\frac{1}{2}}$$

for the prefilter and

$$20 \text{ LOG } |A(z)| = 20 \text{ LOG } (A(z) \cdot A^*(z))^{\frac{1}{2}}$$

for the lattice filter, where

$$A(z) = 1 + \sum_{k=1}^{\text{NPOLE}} A_k e^{-j\omega k}$$

The frequency range of the plots is 0-5000 Hz.

The output power spectrum is computed (4) for the stochastic case from $10 \text{ LOG } P(\omega)$ where

$$P(\omega) = P_{in}(\omega) \cdot |H(\omega)|^2 = P_{in}(\omega) \cdot \frac{1}{A(z) \cdot A^*(z)}$$

For the prefilter, $P_{in}(\omega) = \sigma_{in}^2 = \sigma_v^2$. For the lattice filter, $P_{in}(\omega)$ will be $P_{out}(\omega)$ from the prefilter, and $P(\omega) = P_{in}(\omega) \cdot |H(\omega)|^2 = P_{in}(\omega) \cdot A(z) \cdot A^*(z)$. Again, the frequency range of the plots is 0-5000 Hz.

LSL.FOR

Figure 1 shows the structure of the least squares complex adaptive lattice. Table I lists the lattice parameters and their relationships to the text equations in (3) and to the FORTRAN code in (1). The algorithm from (1) has been modified (from RATFOR) for use by the ARL VAX/VMS computer. Users are referred to (2) for a detailed explanation of filter parameters. The interactive user inputs to program LSL are SNR, desired signal to noise ratio in dB; ALSL, value of α_{CLSL} ; P, order of the filter; and LIM, the desired number of iterations. After LIM number of iterations, the forward predictor coefficients become the filter coefficients of the inverse filter. The convergence properties of this lattice structure have been studied by Hodgkiss (2,3) and are not considered here. The maximum allowable number of iterations is set at 5000 which allows the lattice more than enough time to adapt.

Subroutine ZCPOLY solves the filter polynomial to make sure the roots are outside the unit circle. If a root is inside the unit circle, it is inverted and a new polynomial is synthesized with subroutine SYNTH1. The coefficients are then stored on disk for use by program FILTER.

The forward and backward reflection (PARCOR) coefficients are also stored on disk for possible future use. These parameters are related to the reflection coefficients of an acoustic tube model of the vocal tract. The next step in this research effort will involve making use of these coefficients to directly identify the prefilter transfer function. Table II lists the prefilter coefficients, final lattice coefficients, and both sets of final PARCOR coefficients for the four sample cases presented in Figures 3-6.

The desired signal to noise ratio is achieved by scaling down the added noise (of user specified standard deviation from GEN.FOR) rather than scaling up the signal. This is accomplished by solving for the desired σ of the noise (8):

$$SNR = 10 \text{ LOG } \frac{\text{LIM} \sum_{n=1} \text{sig}^2(n)}{\text{LIM} \sum_{n=1} \text{noise}^2(n)}$$

taking the antilog:

$$10^{(SNR/10)} = \frac{\text{LIM} \sum_{n=1} \text{sig}^2(n)}{\text{LIM} \sum_{n=1} \text{noise}^2(n)}$$

$$\text{LIM} \sum_{n=1} \text{noise}^2(n) = \frac{\text{LIM} \sum_{n=1} \text{sig}^2(n)}{10^{(SNR/10)}}$$

$$\frac{\text{LIM} \sum_{n=1} \text{noise}^2(n)}{\text{LIM}} = \sigma_{\text{noise}}^2 = \frac{\text{LIM} \sum_{n=1} \text{sig}^2(n)}{\text{LIM} \cdot 10^{(SNR/10)}}$$

$$\sigma_{\text{noise}} = \left(\frac{\text{LIM} \sum_{n=1} \text{sig}^2(n)}{\text{LIM} \cdot 10^{(SNR/10)}} \right)^{\frac{1}{2}}$$

Therefore, multiply the noise samples by the factor $\frac{\sigma_{\text{noise}}}{\sigma_{\text{input}}}$.

ACKNOWLEDGMENTS

The complex least squares lattice algorithm in program LSL is an adaptation of a program from the package developed by W. S. Hodgkiss, D. Alexandrou, and J. A. Presley (1-3,9).

The authors would also like to thank J. R. Sacha for the benefit of his generous help and limitless expertise.

REFERENCES

1. D. Alexandrou and W. S. Hodgkiss, "Complex adaptive one-step linear predictor algorithms," MPL TM-341, Marine Physical Laboratory, Scripps Institution of Oceanography, San Diego, CA, 23 June 1982.
2. W. S. Hodgkiss and J. A. Presley, "Adaptive tracking of multiple sinusoids whose power levels are widely separated," IEEE Trans. Acoust., Speech and Signal Processing, Vol. ASSP-29, pp. 710-721, June 1981.
3. W. S. Hodgkiss and J. A. Presley, "The complex adaptive least-squares lattice," IEEE Trans. Acoust., Speech and Signal Processing, Vol. ASSP-30, pp. 330-333, April 1982.
4. L. J. Griffiths, "Rapid measurement of digital instantaneous frequency," IEEE Trans. Acoust., Speech and Signal Processing, Vol. ASSP-23, pp. 207-222, April 1975.
5. L. R. Rabiner and R. W. Schafer, Digital Processing of Speech Signals, Englewood Cliffs, NJ, Prentice-Hall, 1978.
6. B. Friedlander and S. Maitra, "Speech deconvolution by recursive ARMA lattice filters," Proc. 1981 Conference on Acoustics, Speech and Signal Processing, Atlanta, Georgia, March 30 - April 1, 1981, pp. 865-868.

7. A. R. Gray, Jr., and J. D. Markel, "A spectral-flatness measure for studying the autocorrelation method of linear prediction of speech analysis," IEEE Trans. Acoustics, Speech and Signal Processing, Vol. ASSP-22, pp. 207-217, June 1974.
8. B. Friedlander, "System identification techniques for adaptive signal processing," Circuits Systems Signal Process., Vol. 1, No. 1, pp. 3-41, 1982.
9. W. S. Hodgkiss, Personal communication, 8 July 1982.

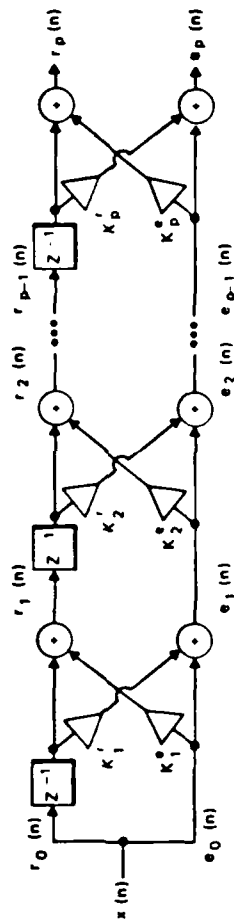
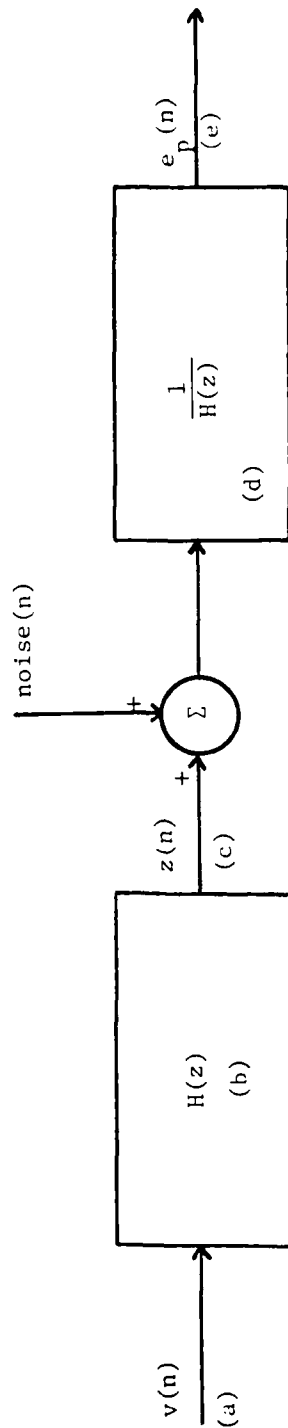


FIGURE 1. LATTICE (INVERSE) FILTER STRUCTURE



- (a) Input to the prefilter, $v(n)$ (impulse sequence or white noise sequence).
- (b) Prefilter transfer function, $20 \log |H(z)|$.
- (c) Output of the prefilter, $z(n)$.
- (d) Lattice (inverse) filter transfer function, $20 \log \left| \frac{1}{H(z)} \right|$.
- (e) Output of the lattice filter (error signal, $e_p(n)$, for deterministic case; output power spectrum, $10 \log P(\omega)$, for stochastic case).

FIGURE 2. BLOCK FLOW DIAGRAM OF LATTICE (INVERSE) FILTER IMPLEMENTATION

November 12, 1982
BAC-LHS:cae

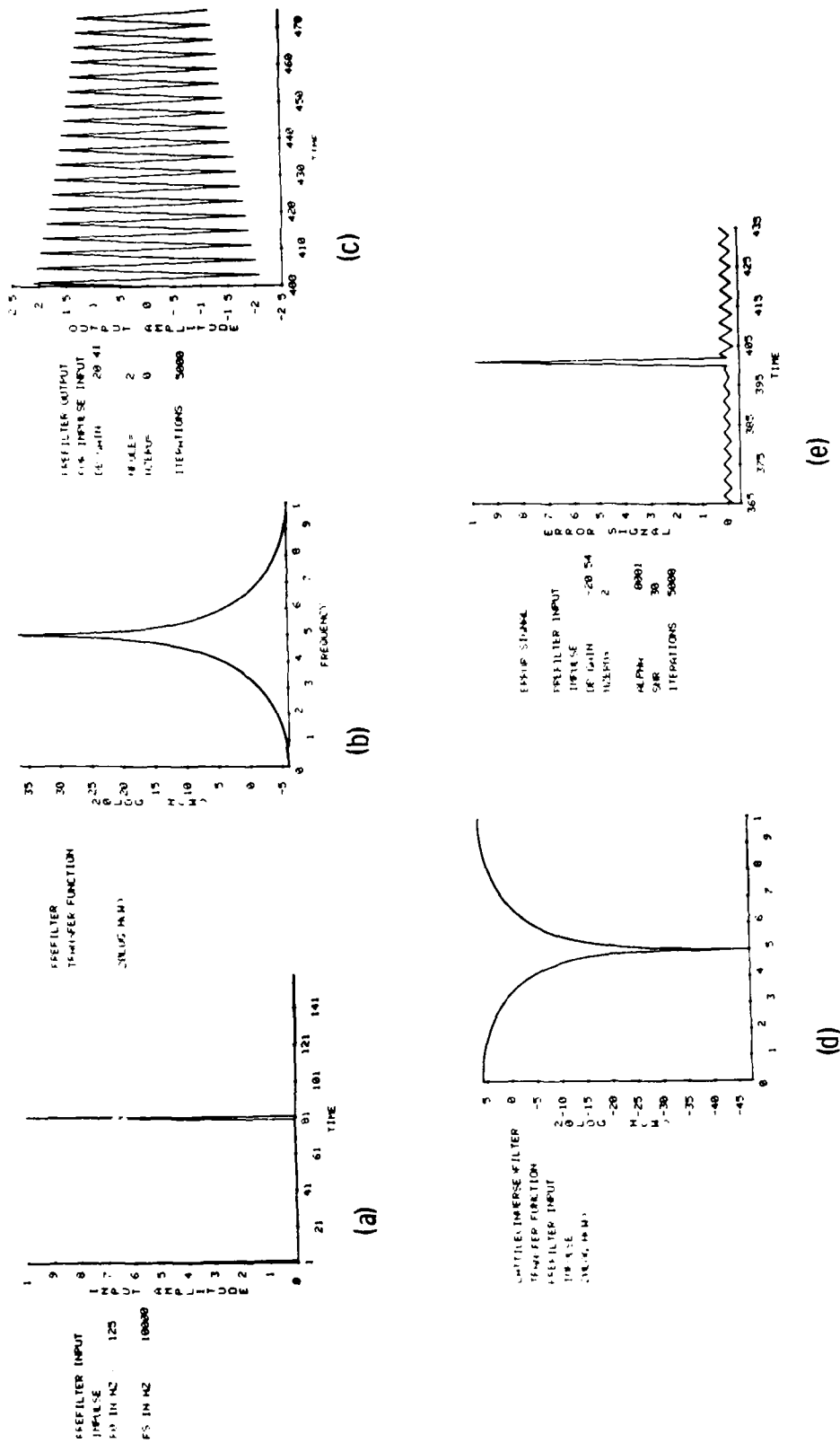


Fig. 3. Each state of the system (see Fig. 2) for an impulse function input to the one-formant case.

November 12, 1982
BAC-LHS:cae

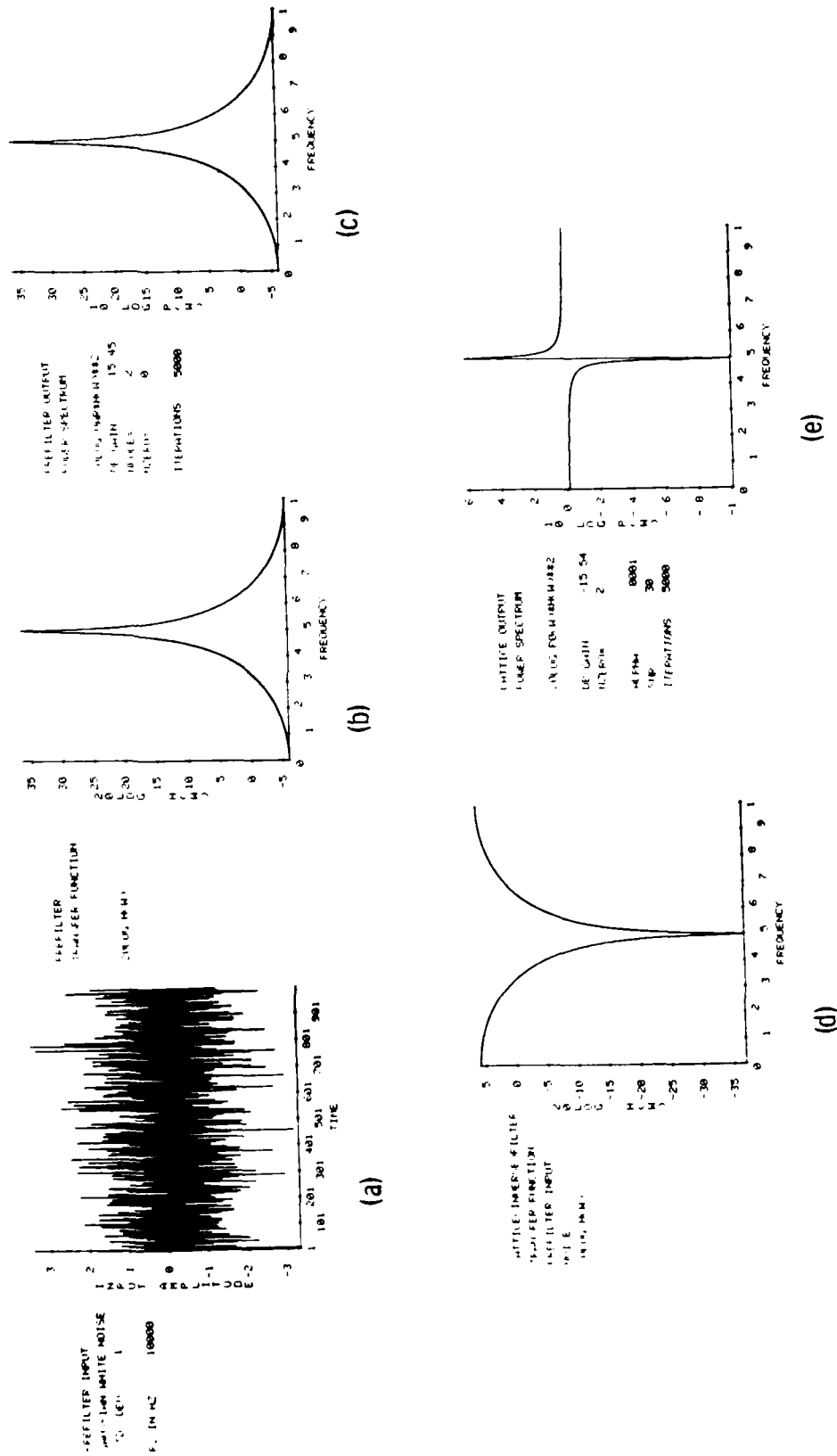


Fig. 4. Each state of the system (see Fig. 2) for a noise input to the one-formant case.

November 12, 1982
BAC-LHS:cae

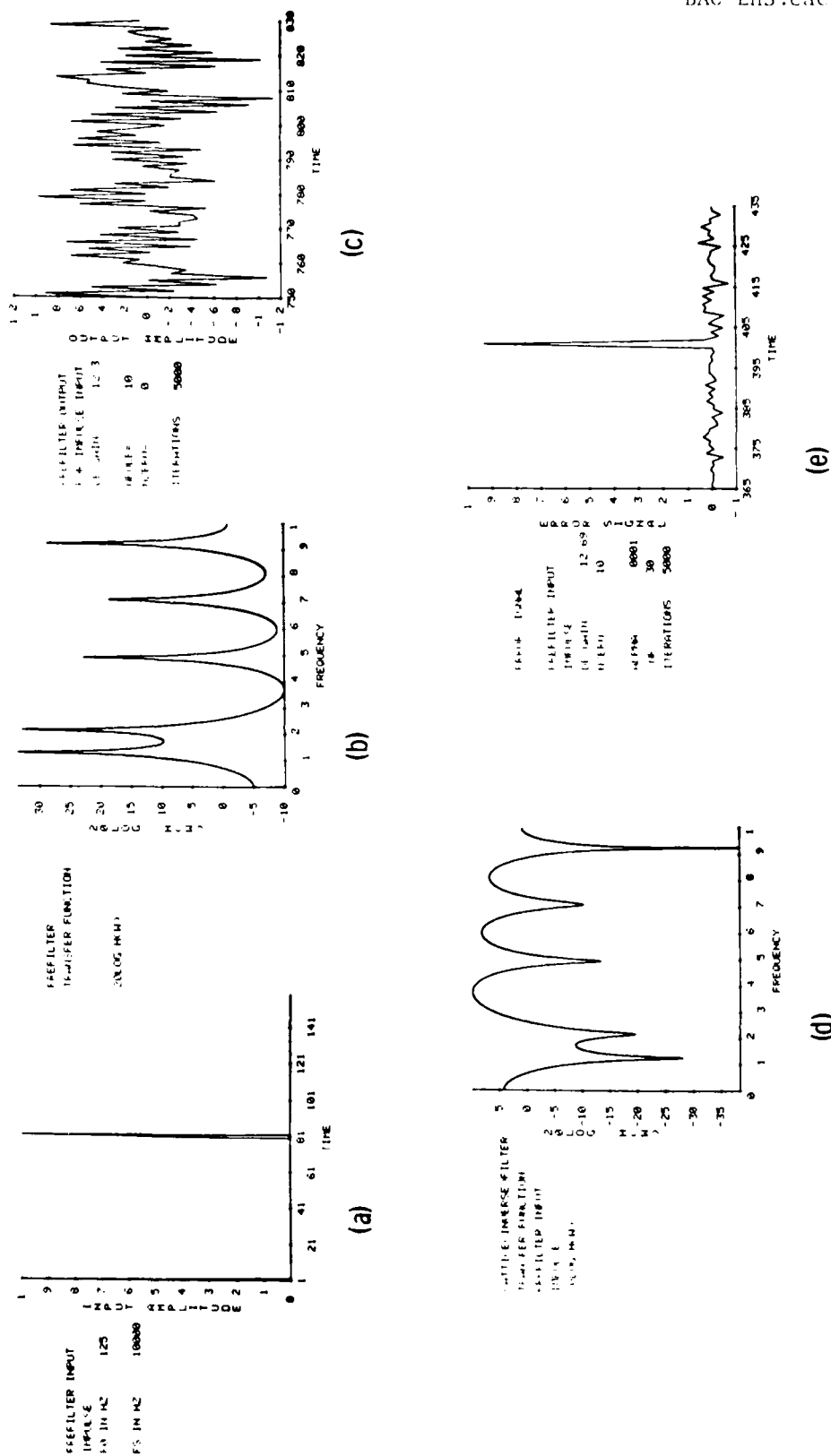
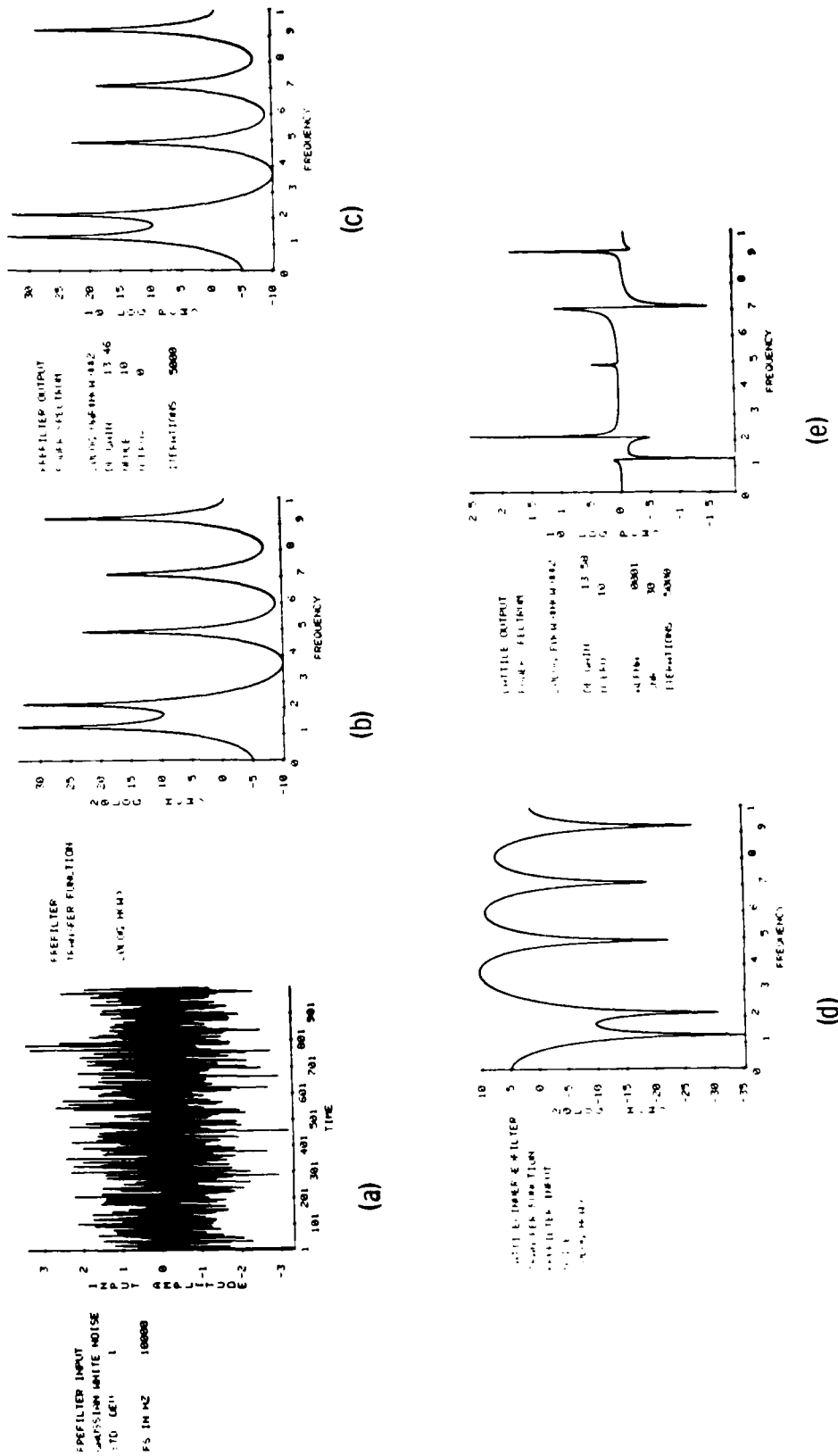


Fig. 5. Each state of the system (see Fig. 2) for an impulse function input to the five-formant case.

November 12, 1982
BAC-LHS:enc



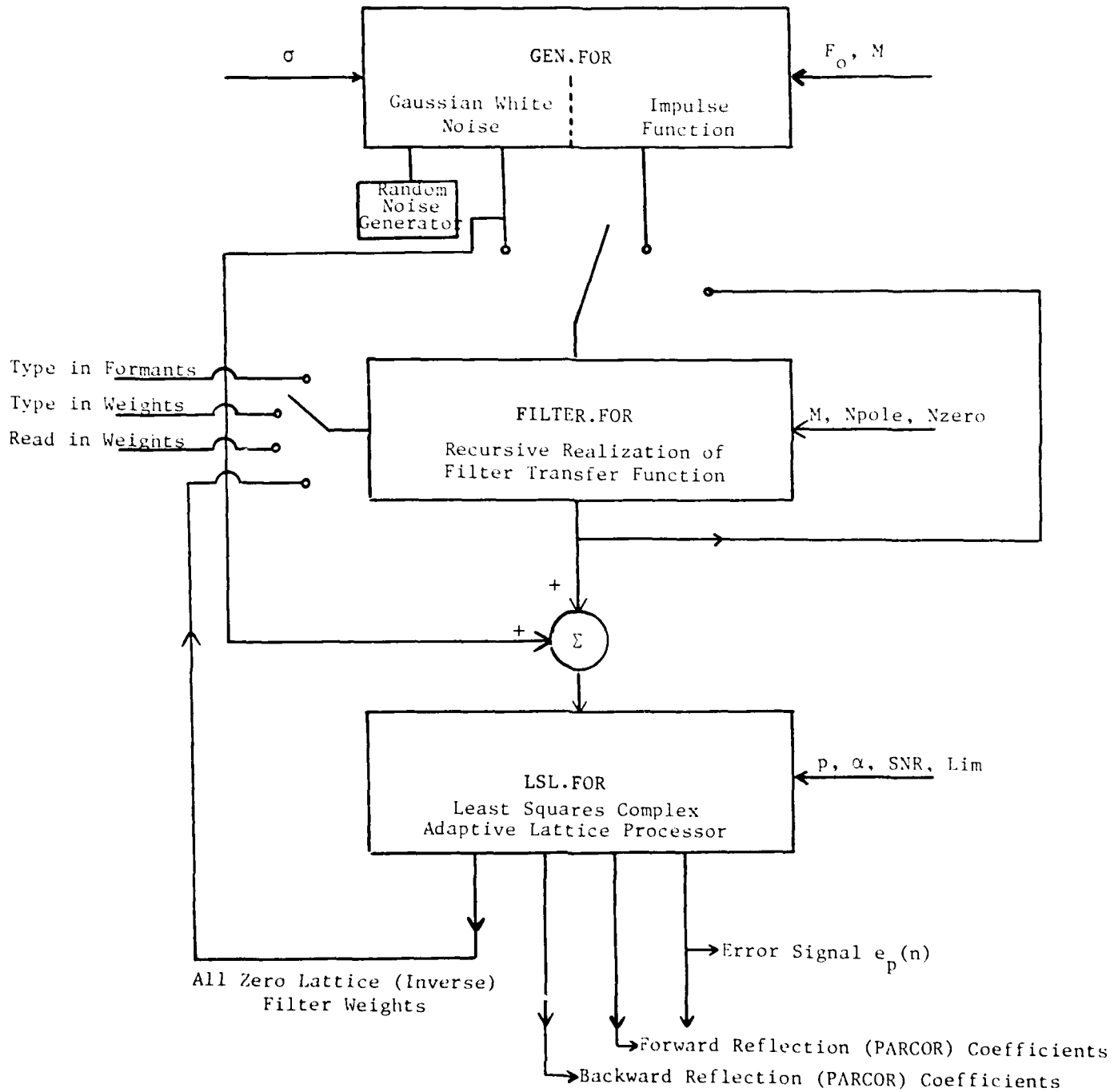


FIGURE 7. CALLING HIERARCHY AND STRUCTURE OF COMMAND PROCEDURE

TABLE I
RELATIONSHIPS BETWEEN LATTICE PARAMETERS, REF. 1 SYMBOLS, AND FORTRAN VARIABLES

Lattice (Inverse) Filter Parameter	REF. 1 SYMBOL	FORTRAN VARIABLE
Vector of forward predictor coefficients at stage i	$a_k^{(i)}(n)$ $1 \leq k \leq i-1$	A(M) $1 \leq M \leq J-1$
Highest (ith) forward predictor at state i	$a_i^{(i)}(n)$	A(J)
Vector of backward predictor coefficients at stage i	$b_k^{(i)}(n)$ $1 \leq k \leq i-1$	B(M,J) $1 \leq M \leq J-1$
Vector of backward predictors at previous stage, previous time	$b_{k-1}^{(i-1)}(n-1)$	B1(M-1,J-1)
Initialize backward predictor at stage i	$b_o^{(i)}(n)$	B(2,J)
Zero out vector of backward predictors for lower stages at time = -1	$b_k^{(i)}(-1)$ $0 \leq k \leq i-1$, $i \neq p$	B1(I,J)
Forward power at stage i	$E_i^e(n)$	EE(J)
Initialize forward power at time = n	$E_o^e(n)$	EE(2)
Backward power at stage i	$E_i^r(n)$	ER(J)
Initialize backward power for lower stages at time = -1	$E_i^r(-1)$ $i \neq p$	ER1(I)
Initialize backward power at time = n	$E_o^r(n)$	ER(2)
Forward reflection coefficient at stage i	$K_i^o(n)$	KE(J)
Backward reflection coefficient at stage i	$K_i^r(n)$	KR(J)
Forward error at stage i	$e_i(n)$	SE(J)
Initialize forward error at time = n	$e_o(n)$	SE(2)
Backward error at stage i	$r_i(n)$	SR(J)
Initialize backward error at time = n	$r_o(n)$	SR(2)
Initialize backward error for lower stages at time = -1	$r_i(-1)$ $i \neq p$	SRI(I)

TABLE 1 (Continued)
RELATIONSHIPS BETWEEN LATTICE PARAMETERS, REF. 1 SYMBOLS, AND FORTRAN VARIABLES

Lattice (Inverse) Filter Parameter	REF. 1 SYMBOL	FORTRAN VARIABLE
Stepsize parameter at stage i	$\gamma_i(n)$	DEL(J)
Initialize stepsize for each stage i at time = -1	$\gamma_i(-1) \quad i \neq 0$	DEL(1)
Input time series	$x(n)$	X(1)
Gain related parameter (gain = $(1 - \gamma_{i-2}(n-1))^{-1}$)	$\gamma_{i-2}(n-1)$	GAM(I-2)
Initialize gamma for time = n-1	$\gamma_{-1}(n-1)$	GAM(1)
Stage variable, current stage	i	J $2 \leq J \leq P+2$
Stage variable, lower stages	k	M $3 \leq M \leq J-1$
Time variable	(n)	Implicit
Time = -1, initial time	(-1)	Denoted by 1(1)
Previous time	(n-1)	Denoted by 1()
Related to fade factor, $(1 - \alpha_{CLS})$	α_{CLS}	ALSL
Filter order	P	P(in ISL), NZERO(in filter)
Number of time samples (iterations)	N	LIN(in ISL), M(in filter)

Case Figure No.	Prefilter Formant, Hz	Prefilter Bandwidth, Hz	Prefilter A_k , $k=1, \dots, N_{\text{pole}}$	Lattice Filter A_k , $k=1, \dots, N_{\text{zero}}$
3	2500.00	150.00	0.000000 0.985112	0.000000 0.995760
4	2500.00	150.00	0.000000 0.985112	-0.002642 0.985765
5	650.30 1075.70 2463.10 3558.30 4631.30	94.10 91.40 107.40 198.70 89.80	-0.266366 -0.522137 0.174152 0.247624 0.513600 0.260513 0.141718 -0.496971 -0.241914 0.943518	-0.232250 -0.543238 0.145636 0.211085 0.515390 0.260130 0.067095 -0.420906 -0.221051 0.860902
6	650.30 1075.70 2463.10 3558.30 4631.30	94.10 91.40 107.40 198.70 89.80	-0.266366 -0.522137 0.174152 0.247624 0.513600 0.260513 0.141718 -0.496971 -0.241914 0.943518	-0.263710 -0.529765 0.172208 0.254260 0.508698 0.264767 0.146636 -0.505169 -0.236822 0.941123

TABLE II

FORMANT FREQUENCIES, THEIR BANDWIDTHS, AND CORRESPONDING
FILTER COEFFICIENTS FOR THE FOUR INPUT CASES DEPICTED IN FIGS. 3-6

Command Procedure LATTICE.COM
Controls Sequence of Program Execution for VT100 Terminals

[illegible]

***** TO STOP SCROLLING, PULL DOWN SCROLL BAR *****

REAR. CHARGES: 1 EASY SQUARES ACCTIVE ALGEBRA AND PRINTING PACKAGE

0474 A. C111.0222

THIS PROGRAM WAS BEING OFFERED AS OF 12/6/42. THE LISTINGS SUBMITTED BY THE BUREAU OF INVESTIGATION ON 12/12/42 ARE NO LONGER CURRENT. THE LISTINGS FOR THE FOLLOWING PROGRAMS: FLETCHER, GUN, FOR, GUN, AND CIVILIAN.

3. CHAIRMAN CONCERNED ABOUT THE COMPLETING, TAKING AND EXECUTION OF EACH PHASE. THE CHAIRMAN SUGGESTED WE AGREE FOR EACH PHASE TO BE FULL.

THE FOLLOWING INFORMATION IS FOR YOUR INFORMATION ONLY. IT IS NOT TO BE USED FOR ANY OTHER PURPOSE. IT IS NOT TO BE REPRODUCED OR TRANSMITTED IN ANY FORM OR BY ANY MEANS, ELECTRONIC OR MECHANICAL, INCLUDING PHOTOCOPYING, RECORDING, OR BY ANY INFORMATION STORAGE AND RETRIEVAL SYSTEM. IT IS NOT TO BE USED FOR ANY OTHER PURPOSE. IT IS NOT TO BE REPRODUCED OR TRANSMITTED IN ANY FORM OR BY ANY MEANS, ELECTRONIC OR MECHANICAL, INCLUDING PHOTOCOPYING, RECORDING, OR BY ANY INFORMATION STORAGE AND RETRIEVAL SYSTEM.

TO EIGHT. LOTS HAVE 15 FIRST PURCHASES WILL BE LIKE TO DIE.

ERATED YOU ENJOYED MOVING AT A PLOT ON THE SCREEN, PRESS THE SPACE BAR TO
ERASE THE PLOT TO ERASE AND THE NEXT PLOT TO APPEAR.

WHEN USING THE TEXTOLVY SYSTEM, YOU WILL NEED TO REPESS THE RETURN BUTTON THREE TIMES TO GET THE CLOSING SPACE.

PLEASE WAIT FOR THE COMPUTER TO COMPILE AND LINK THE PROGRAMS. YOU WILL RECEIVE A MESSAGE WHEN THE COMPUTER IS READY.

[illegible]

THIS NUMBER WAS BEEN RECEIVED AS OF 176042. THE LISTINGS SUPPLIED
WITH THE TELETYPE MESSAGE NO 82-219, DATED 11/17/82 ARE THE LONGER
C CORDACT. PLEASE NOTIFY NEW LISTINGS.

THIS PROGRAM GENERATES COMPLEX GAUSSIAN NOISE WITH VARIABLE STANDARD DEVIATION AND COMPLEX INDEPENDENT SEQUENCE OF VARIABLE FREQUENCY.

(U.S. 251107' (U.S. 1810000

```

10000 TYPE, STATED NUMBER OF SAMPLES TO GENERATE.
      TYPE, STATED NUMBER SAMPLE CORRESPOND TO DESIGN NUMBER OF
      1 ITERATION OF LATTICE.
      TYPE, 10 (1000000000).
      ACCELO, 0

```

BYE, "LETTER TESTED STANDARD DEVIATION OF GAUSSIAN WHITE NOISE."
THEY, "LETTER TESTED STANDARD DEVIATION OF GAUSSIAN WHITE NOISE."

```

        NUMBER, *NEW, NEWEN, SPREADSHEET OF IMPULSE FUNCTION IN M7.
      C(20), *NEWEN
      NUMBER = EXP(NAT(NUMBER))
      EN = INT(NUMBER) / NUNBER
      PRINT FILE NUMBER.
    ELSE=2

```

CLTYPE(0)E16,600) "FILE
NUMBER(S),ALPHABETIC INFORMATION IS STORED IN DISK FILE: ,15)

[illegible]

Recursive Realization of Prefilter and Lattice (Inverse) Filter Transfer Functions

THE FOLLOWING TABLES SHOW THE PERCENTAGE OF THE TOTAL POPULATION OF THE UNITED STATES IN THE SEVERAL RACES AND NATIONALITIES IN 1900 AND 1910.

[illegible]

THESE RESULTS ARE IN FULL AGREEMENT WITH THE PREVIOUS RESULTS.


```

WRITE(UNIT=600)DATA
CLOSE(UNIT=600)
WRITE(UNIT=600)UNIT
WRITE(UNIT=600)LCASE
WRITE(UNIT=600)TYPE
WRITE(UNIT=600)INFO
POLARIZATION, SOURCE INFORMATION IS STORED IN DISK FILE: (15)

```

[illegible][illegible][illegible]

15/11/6114.06.110200 107

7. CHARACTERISTIC OUTPUT OF $\text{PREF}(1,1,1,1,1)$
 FOR $\text{PREF}(1,1,1,1,1)$, $\text{PREF}(1,1,1,1,1)$
 IS $\text{PREF}(1,1,1,1,1)$

THE DISSEMINATION OF INFORMATION IS A RESPONSIBILITY, AND WE WILL MEET.

DEAR (05115,000) (VVV(1),1=1,1,1M)
VVVV(5V,45|2,4)

10-10-68 11:00AM 917VA 20-21-68

```

1  STH=0.0
2  DO 100 I=1,N
3     STH=STH+(A(I)*C)/D
4     CONTINUE
5  STH=STH/100
6  STH=STH*100
7  STH=STH/100
8  STH=STH*100
9  STH=STH/100
10 STH=STH*100

```

THESE RESULTS ARE IN ACCORD WITH THE FINDINGS OF OTHER STUDIES ON THE EFFECTS OF STRESS ON THE IMMUNE SYSTEM.

042/11025-100
0425/0450/11025-11025
915/11002

77.12.00 12
 (11)35(1000000)+1(1000000)1
 11.11.11 11 0
 * 12.01.11 01 1100000000000000

THE FOLLOWING IS TABLE FROM REF. 1 (COMMENTS WITH).

```

THE NEXT SECTION INITIALIZES THE INDUSTRIES AT TIME ZERO FOR ALL STAGES OF
- THE LIFECYCLE. FOR THIS PROGRAM, THE STATES WILL BE COUNTS TO C=COUNT BY MONTH
- IN THE SECTOR. THESE STATEMENTS CORRESPOND TO EQUATIONS A4-A10 IN MY 7.
TODA ACCELED.
```

0	$\mathbf{f}(1,1) = (1,1)$
1	$\mathbf{f}(1,2) = (1,2)$
2	$\mathbf{f}(1,3) = (1,3)$
3	$\mathbf{f}(1,4) = (1,4)$
4	$\mathbf{f}(2,1) = (2,1)$
5	$\mathbf{f}(2,2) = (2,2)$
6	$\mathbf{f}(2,3) = (2,3)$
7	$\mathbf{f}(2,4) = (2,4)$
8	$\mathbf{f}(3,1) = (3,1)$
9	$\mathbf{f}(3,2) = (3,2)$
10	$\mathbf{f}(3,3) = (3,3)$
11	$\mathbf{f}(3,4) = (3,4)$
12	$\mathbf{f}(4,1) = (4,1)$
13	$\mathbf{f}(4,2) = (4,2)$
14	$\mathbf{f}(4,3) = (4,3)$
15	$\mathbf{f}(4,4) = (4,4)$

MULTIPLY $END(1)$ FOR ALL STACKS = $END(1,0) = .001$ TO AVOID DIVIDING
 BY ZERO [$END(1,0) = .4$].

$$\begin{aligned} \psi^{\dagger} \psi &= (1, 1, 1, 1) \\ \psi^{\dagger} \gamma^{\mu} \psi &= (1, 1, 1, 1) \\ \psi^{\dagger} \gamma^{\mu} \gamma^{\nu} \psi &= (1, 1, 1, 1) \\ \psi^{\dagger} \gamma^{\mu} \gamma^{\nu} \gamma^{\rho} \psi &= (1, 1, 1, 1) \\ \psi^{\dagger} \gamma^{\mu} \gamma^{\nu} \gamma^{\rho} \gamma^{\sigma} \psi &= (1, 1, 1, 1) \end{aligned}$$

THE FOUR LOOP IS EXECUTED FOR EACH TWO SAMPLES.
INITIALIZE PARAMETERS AT START ZERO OF THE LATTICE. THE NEXT FOUR STATEMENTS
CORRESPOND TO EQUATIONS 4-7C IN REF. 7.

$$\begin{aligned} \text{for } j=0 \text{ to } n-1 \text{ do} \\ \quad \text{if } (j \bmod 2) = 0 \text{ then} \\ \quad \quad \text{swap}(a[j], a[j+1]) \\ \quad \text{end if} \\ \quad \text{swap}(a[j], a[n-j-1]) \\ \text{end for} \end{aligned}$$

THE NEXT LOOP IS EXECUTED FOR EACH STAGE OF THE LATTICE.
THE NEXT GROUP OF STATEMENTS CORRESPOND TO EQUATIONS 4W-4O.

[illegible]

November 12, 1982
BAC-LHS:cac

SUBROUTINE SYNTH(N,M)

THIS SUBROUTINE SYNTHESIZES A COMPLEX POLYNOMIAL IN Z**+1 FROM AN
ARRAY OF THE POOTS OF THE POLYNOMIAL.

M IS THE ORDER OF THE POLYNOMIAL.
B (OUTPUT) IS AN ARRAY OF THE FILTER COEFFICIENTS OF THE POLYNOMIAL
IN Z**+1.

```

45      COMPLEX A(16),B(17),ANEG(16)
        PI=4.0*ATAN(1.0)
        DO 555 I=1,N
          CONTINUE
        DO 10 K=1,M
          B(K)=(0.0,0.0)
          CONTINUE
        R(2)=-1.0*A(1)
        R(1)=(1.0,0.0)
        DO 20 I=2,N
          DO 10 J=(I+1),2,-1
            R(J)=R(J)-A(I)*R(J-1)
          CONTINUE
        R(1)=A(1)
        CONTINUE
        RETURN
      END

```

SUBROUTINE SYNTH(N,FF,BRW,HR,CC,FSS)

C THIS PROGRAM HAS BEEN REVISED AS OF 1/24/83. THE LISTINGS SUPPLIED
C WITH ARL INTERNAL MEMORANDUM R2-219, DATED 11/12/82 ARE NO LONGER
C CORRECT. PLEASE OBTAIN NEW LISTINGS.
C *****

C THIS SUBROUTINE SYNTHESIZES A COMPLEX POLYNOMIAL IN Z**+1 FROM AN
C ARRAY OF FORMANT FREQUENCIES AND AN ARRAY OF THEIR RESPECTIVE BANDWIDTHS.
C N IS THE NUMBER OF FORMANTS AND THE DEGREE OF THE POLYNOMIAL.
C FF IS AN ARRAY OF THE FORMANT FREQUENCIES IN HZ.
C BRW IS AN ARRAY OF THEIR BANDWIDTHS IN HZ.
C HR (OUTPUT) IS AN ARRAY OF THE REAL PARTS OF THE FILTER COEFFICIENTS.
C CC (OUTPUT) IS AN ARRAY OF THE IMAGINARY PARTS OF THE FILTER COEFFICIENTS.

C SEE TEXT FOR RELATIONSHIPS BETWEEN FORMANTS, THEIR BANDWIDTHS,
C AND THE FILTER COEFFICIENTS.

```

      COMPLEX A(16),B(17),ANEG(16),PFXP,ANEXP,F2P,E2N
      REAL HR(17),CC(17),FF(R),BRW(R)
      PI=4.0*ATAN(1.0)
      DO 107 I=1,(N/2)
        FF(I)=FF(I)/FSS
        BRW(I)=BRW(I)/FSS
        E1=EXP(-PI*BRW(I)*PI)
        PFXP=CMPLX(0.0,(2.0*PI*FF(I)))
        ANEXP=CMPLX(0.0,(-2.0*PI*FF(I)))
        F2P=CEXP(PFXP)
        E2N=CEXP(ANEXP)
      C THESE ARE THE ZPLANE POLES.
      A(I*2)=F1*E2P
      A(I*2-1)=F1*E2N
      CONTINUE
      DO 30 K=1,17
        R(K)=(0.0,0.0)
      CONTINUE
      R(2)=-1.0*A(1)
      R(1)=(1.0,0.0)
      DO 20 I=2,N
        DO 10 J=(I+1),2,-1
          R(J)=R(J)-A(I)*R(J-1)
        CONTINUE
      CONTINUE
      DO 92 MW=1,N
        R(MW)=R(MW)*R(MW+1)
        CC(MW)=A(MW)*R(MW+1)
      CONTINUE
      RETURN
      END

```

November 12, 1982
BAC-LHS:cac

```

SUBROUTINE ADDRESS(ARRAY, IARRAY)
*
C THIS SUBROUTINE FINDS THE ARRAY CONTAINING THE ELEMENT WITH THE MAXIMUM
C VALUE.
C ARRAY IS AN ARRAY OF POINTS.
C IARRAY IS THE INDEX OF THE ARRAY.
*
      DIMENSION IARRAY(1000)
      ADDRESS=0
      IADDRESS=0
      DO 10 I=1, IARRAY
        IF (ARRAY(I)) > ADDRESS THEN
          ADDRESS=ARRAY(I)
          IADDRESS=I
        END IF
      CONTINUE
      RETURN
      END

```

```

SUBROUTINE ADDRESS(ARRAY, IARRAY)
*
C THIS SUBROUTINE FINDS THE MAXIMUM VALUE IN THE ARRAY.
C ARRAY IS AN ARRAY OF POINTS.
C IARRAY IS THE INDEX OF THE ARRAY.
*
      DIMENSION ADDRESS(1000)
      ADDRESS=0
      DO 10 I=1, IARRAY
        IF (ARRAY(I)) > ADDRESS THEN
          ADDRESS=ARRAY(I)
        END IF
      CONTINUE
      RETURN
      END

```

```

SUBROUTINE ADDRESS(ARRAY, IARRAY)
*
C THIS SUBROUTINE FINDS THE MAXIMUM VALUE IN THE ARRAY.
C ARRAY IS AN ARRAY OF POINTS.
C IARRAY IS THE INDEX OF THE ARRAY.
*
      DIMENSION ADDRESS(1000)
      ADDRESS=0
      DO 10 I=1, IARRAY
        IF (ARRAY(I)) > ADDRESS THEN
          ADDRESS=ARRAY(I)
        END IF
      CONTINUE
      RETURN
      END

```

```

SUBROUTINE ADDRESS(ARRAY, IARRAY)
*
C THIS SUBROUTINE FINDS THE MAXIMUM VALUE IN THE ARRAY.
C ARRAY IS AN ARRAY OF POINTS.
C IARRAY IS THE INDEX OF THE ARRAY.
*
      DIMENSION ADDRESS(1000)
      ADDRESS=0
      DO 10 I=1, IARRAY
        IF (ARRAY(I)) > ADDRESS THEN
          ADDRESS=ARRAY(I)
        END IF
      CONTINUE
      RETURN
      END

```

APPENDIX B

SAMPLE TERMINAL SESSION FOR THE CASE ILLUSTRATED IN FIG. 3

```

LATTICE
WHICH PROGRAM TO USE?: GEN
$ENTER NUMBER OF SAMPLES TO GENERATE.
$THIS NUMBER SHOULD CORRESPOND TO DESIRED NUMBER OF ITERATIONS OF LATTICE.

$(C1<=NK=5000).
5000
$ENTER DESIRED STANDARD DEVIATION OF GAUSSIAN WHITE NOISE.
$(MUST BE FLOATING POINT).
1.0
$ENTER DESIRED FREQUENCY OF IMPULSE FUNCTION IN HZ.
125.0
$WHITE NOISE IN FILE 1.
$GAUSSIAN NOISE IN FILE 2.
$IMPULSE INPUT IN FILE 3.
$MAKE SURE TO RECORD THESE NUMBERS.
FORTRAN STOP
ENTER Y/YES TO CONTINUE: Y
WHICH PROGRAM TO USE?: FILTER
$ENTER 1 IF YOU WOULD LIKE TO READ CALCULATED INVERSE FILTER COEFFICIENTS
FROM
DISK.
$ENTER 0 IF YOU WOULD LIKE TO TYPE IN FILTER COEFFICIENTS.
$ENTER 2 IF YOU WOULD LIKE TO SPECIFY FORMANTS AND BANDWIDTHS.
$ENTER 3 IF YOU WOULD LIKE TO READ IN PREFILTER COEFFICIENTS FROM DISK.
2
$ENTER NUMBER OF SAMPLES TO ANALYZE.
$THIS NUMBER SHOULD CORRESPOND TO DESIRED NUMBER OF ITERATIONS OF THE LATTICE.
ICE (M
<=5000).
5000

```

November 12, 1982
BAC-LHS:cac

\$IF YOU WISH TO SPECIFY FORMANTS AND BANDWIDTHS, THERE WILL BE TWO (COMPLE
N CONJ
UGATED) POLES FOR EACH FORMANT.
\$ENTER NUMBER OF POLES (1<=NPOLE<=16).

I= 1
\$ENTER FORMANT FREQUENCY IN HZ.
2500.0
\$ENTER BANDWIDTH OF THIS FORMANT FREQUENCY IN HZ.
150.0
\$FILTER COEFFICIENTS IN ORDER OF INCREASING POWER.
\$CAO=1.0).

0.000000 0.000000
0.985112 0.000000

\$ENTER 1 IF YOU WOULD LIKE TO SAVE THIS INFORMATION ON DISK.
\$OTHERWISE, ENTER 0.

1
\$ENTER NUMBER OF DISK FILE TO SAVE THIS IN.
\$USE A FILE NUMBER BETWEEN 40 AND 60, INCLUSIVE.
40

(SCREEN WILL ERASE)

THIS INFORMATION STORED IN DISK FILE 40
 PREFILTER BANDWIDTH
 FORMANT 150.00
 2500.00
 FILTER COEFFICIENTS (INCREASING POWER, $AB=1$).
 0.000000 0.000000
 0.985112 0.000000
 #ENTER DESIRED INPUT FILE NUMBER.
 #NOISE INPUT FILE 2
 #IMPULSE INPUT FILE 3

OUTPUT IN FILE: 21
 #BE SURE TO RECORD THIS NUMBER.
 #ENTER 1 IF YOU WOULD LIKE TO PLOT INPUT TIME SERIES.
 #ENTER 0 OTHERWISE.
 1
 #ENTER FIRST AND LAST TIME POINTS OF INPUT TO PLOT.
 (1<=M<=) 5000
 1
 149
 #ENTER INTERVAL BETWEEN POINTS ON PLOT.
 1

(PLOT (a) APPEARS ON SCREEN)

(DEPRESS CR OR SPACE BAR ONCE OR TWICE TO ERASE PLOT)

#ENTER 1 IF YOU WOULD LIKE TO PLOT THE INPUT AGAIN ON A DIFFERENT SCALE.
 #OTHERWISE, TYPE IN 0.

0
 #ENTER 1 IF YOU WOULD LIKE TO PLOT OUTPUT TIME SERIES.
 #OTHERWISE ENTER 0.

1
 #ENTER FIRST AND LAST TIME POINTS OF OUTPUT TO PLOT.
 (1<=M<=) 5000

400
 400

#ENTER INTERVAL BETWEEN POINTS ON PLOT.

1
 (PLOT (c) APPEARS ON SCREEN)

#ENTER 1 IF YOU WOULD LIKE TO PLOT THE OUTPUT AGAIN ON A DIFFERENT SCALE.
#OTHERWISE, TYPE IN 0.
0
#ENTER 1 IF YOU WOULD LIKE TO PLOT TRANSFER FUNCTION.
#OTHERWISE ENTER 0.
1

(PLOT (b) APPEARS ON SCREEN)

FORTRAN STOP
ENTER YES TO CONTINUE: Y
WHICH PROGRAM TO USE? LSL
ENTER FILTER ORDER (1<=P<=16):
2
ENTER ALPHA(CLSL):
NO ANALYSIS(1.0).
0001
SPECIFY DESIRED SNR IN DB AS A REAL NUMBER.
20.0
ENTER NUMBER OF ITERATIONS DESIRED.
1 (1<=LIM<=5000).
5000

FORWARD PREDICTOR COEFFICIENTS IN FILE: 12
BACKWARD PREDICTOR COEFFICIENTS IN FILE: 13
FORWARD PARCOR COEFFICIENTS IN FILE: 14
BACKWARD PARCOR COEFFICIENTS IN FILE: 15
FINAL FORWARD PREDICTOR COEFFICIENTS IN FILE: 11
ERROR SIGNAL IN FILE: 16
FINAL FORWARD PARCOR COEFFICIENTS IN FILE: 17
FINAL BACKWARD PARCOR COEFFICIENTS IN FILE: 18
MAKE SURE TO RECORD THESE NUMBERS.
IF USING THE TEKTRONIX TERMINAL, DEPRESS PAGE BUTTON
TO CLEAR SCREEN FOR HARD COPY. (DON'T DO THIS YET)
ENTER NUMBER OF FILE TO SAVE COEFFICIENTS IN.
PLEASE USE A FILE NUMBER BETWEEN 70 AND 99, INCLUSIVE.
70

(PRESS PAGE BUTTON NOW TO CLEAR TEKTRONIX SCREEN)

```

FILTER FOR PREFILTER INPUT=IMPULSE
FINAL BACKWARD PARCOR COEFFICIENTS
  0.000000  0.000000
  0.995760  0.000000
FINAL FORWARD PARCOR COEFFICIENTS
  0.000000  0.000000
  0.995458  0.000000
FORWARD PREDICTOR COEFFICIENTS:
  0.000000  0.000000
  0.995760  0.000000
2-1PLANE ZEROES (ABS. VALUE):
  1.002127
  1.002127
FORTRAN STOP
ENTER 'YES' TO CONTINUE: Y
WHICH PROGRAM TO USE?: FILTER
#ENTER 1 IF YOU WOULD LIKE TO READ CALCULATED INVERSE FILTER COEFFICIENTS
FROM DISK.
#ENTER 0 IF YOU WOULD LIKE TO TYPE IN FILTER COEFFICIENTS.
#ENTER 2 IF YOU WOULD LIKE TO SPECIFY FORMANTS AND BANDWIDTHS.
#ENTER 3 IF YOU WOULD LIKE TO READ IN PREFILTER COEFFICIENTS FROM DISK.
1
#ENTER 1 IF YOU WOULD LIKE TO PLOT OUTPUT TIME SERIES.
#OTHERWISE ENTER 0.
1
#ENTER FIRST AND LAST TIME POINTS OF OUTPUT TO PLOT.
  (1<=M<=) 5000
365
435
#ENTER INTERVAL BETWEEN POINTS ON PLOT.
  1

```

(THIS PLOT NOT SHOWN IN FIG. 3; IT IS EQUIVALENT TO PLOT (e))

November 12, 1982
BAC-LHS:cac

#ENTER 1 IF YOU WOULD LIKE TO PLOT THE OUTPUT AGAIN ON A DIFFERENT SCALE.
#OTHERWISE, TYPE IN 0.

0

#ENTER 1 IF YOU WOULD LIKE TO PLOT ERROR SIGNAL.
#OTHERWISE ENTER 0.

1

#ENTER FIRST AND LAST TIME POINTS OF ERROR TO PLOT.
(1<=M<=) 5000

355

435

#ENTER INTERVAL BETWEEN POINTS ON PLOT.
1

(PLOT (e) APPEARS ON SCREEN)

ENTER 1 IF YOU WOULD LIKE TO PLOT ERROR AGAIN ON ANOTHER SCALE.
#OTHERWISE, TYPE IN 0.

0

#ENTER 1 IF YOU WOULD LIKE TO PLOT TRANSFER FUNCTION.
#OTHERWISE ENTER 0.

1

(PLOT (d) APPEARS ON SCREEN)

FORTRAN STOP
ENTER YES] TO CONTINUE: N
ENTER YES] FOR HARD COPY OF ALL FILTER COEFFICIENTS: N
ENTER YES] TO DELETE ALL DATA FILES: N
#

(END OF COMMAND PROCEDURE)

DISTRIBUTION LIST FOR ARL TM 82-219, by B. A. Cooper and L. H. Sibul,
dated November 12, 1982

Commander
Naval Sea Systems Command
Department of the Navy
Washington, DC 20362

Attention: Dr. E. G. Liszka, PMS 406B
Copy No. 1

Attention: Mr. D. Porter, SEA 63R1
Copy No. 2

Attention: Mr. D. C. Houser, SEA 63R-14
Copy No. 3

Attention: Code SEA 9961 (Library)
Copies 4 and 5

Commander
Naval Air Systems Command
Department of the Navy
Washington, DC 20362

Attention: Mr. J. C. Smith
Copy No. 6

Director
Applied Research Laboratory
University of Texas
Austin, TX 78712

Attention: Dr. J. F. Willman
Copy No. 7

Marine Physical Laboratory
Scripps Institute of Oceanography
San Diego, CA 92152

Attention: W. S. Hodgkiss
Copy No. 8

Attention: K. M. Watson
Copy No. 9

Director
Applied Physics Laboratory
University of Washington
Seattle, WA 98105

Attention: Mr. C. Sienkiewicz
Copy No. 10

Commanding Officer
Naval Coastal Systems Center
Panama City, FL 32401

Attention: Dr. David Skinner
Copy No. 11

Bolt Beranek and Newman, Inc.
1300 North 17th Street
Arlington, VA 22209

Attention: Dr. Henry Cox
Copy No. 12

Systems Control Technology, Inc
1801 Page Mill Road
P. O. Box 10180
Palo Alto, CA 94303

Attention: Dr. Benjamin Friedlander
Copy No. 13

Department of Electrical Engineering
College of Engineering and Applied Science
The University of Rochester
Rochester, NY 14627

Attention: Prof. E. L. Titlebaum
Copy No. 14

Defense Technical Information Center
Cameron Station
Alexandria, VA 22314

Copies 15 through 20

**DAI
FILM**